

```

using System;
using Microsoft.EntityFrameworkCore.Migrations;

namespace MyCompany.Migrations
{
    public partial class _initial : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "AspNetRoles",
                columns: table => new
                {
                    Id = table.Column<string>(nullable: false),
                    Name = table.Column<string>(maxLength: 256, nullable: true),
                    NormalizedName = table.Column<string>(maxLength: 256, nullable: true),
                    ConcurrencyStamp = table.Column<string>(nullable: true)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_AspNetRoles", x => x.Id);
                });

            migrationBuilder.CreateTable(
                name: "AspNetUsers",
                columns: table => new
                {
                    Id = table.Column<string>(nullable: false),
                    UserName = table.Column<string>(maxLength: 256, nullable: true),
                    NormalizedUserName = table.Column<string>(maxLength: 256, nullable: true),
                    Email = table.Column<string>(maxLength: 256, nullable: true),
                    NormalizedEmail = table.Column<string>(maxLength: 256, nullable: true),
                    EmailConfirmed = table.Column<bool>(nullable: false),
                    PasswordHash = table.Column<string>(nullable: true),
                    SecurityStamp = table.Column<string>(nullable: true),
                    ConcurrencyStamp = table.Column<string>(nullable: true),
                    PhoneNumber = table.Column<string>(nullable: true),
                    PhoneNumberConfirmed = table.Column<bool>(nullable: false),
                    TwoFactorEnabled = table.Column<bool>(nullable: false),
                    LockoutEnd = table.Column<DateTimeOffset>(nullable: true),
                    LockoutEnabled = table.Column<bool>(nullable: false),
                    AccessFailedCount = table.Column<int>(nullable: false)
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_AspNetUsers", x => x.Id);
                });

```

```

migrationBuilder.CreateTable(
    name: "ServiceItems",
    columns: table => new
    {
        Id = table.Column<Guid>(nullable: false),
        TitleImagePath = table.Column<string>(nullable: true),
        MetaTitle = table.Column<string>(nullable: true),
        MetaDescription = table.Column<string>(nullable: true),
        MetaKeywords = table.Column<string>(nullable: true),
        DateAdded = table.Column<DateTime>(nullable: false),
        Title = table.Column<string>(nullable: false),
        Subtitle = table.Column<string>(nullable: true),
        Text = table.Column<string>(nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_ServiceItems", x => x.Id);
    });

```

```

migrationBuilder.CreateTable(
    name: "TextFields",
    columns: table => new
    {
        Id = table.Column<Guid>(nullable: false),
        Subtitle = table.Column<string>(nullable: true),
        TitleImagePath = table.Column<string>(nullable: true),
        MetaTitle = table.Column<string>(nullable: true),
        MetaDescription = table.Column<string>(nullable: true),
        MetaKeywords = table.Column<string>(nullable: true),
        DateAdded = table.Column<DateTime>(nullable: false),
        CodeWord = table.Column<string>(nullable: false),
        Title = table.Column<string>(nullable: true),
        Text = table.Column<string>(nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_TextFields", x => x.Id);
    });

```

```

migrationBuilder.CreateTable(
    name: "AspNetRoleClaims",
    columns: table => new
    {
        Id = table.Column<int>(nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        RoleId = table.Column<string>(nullable: false),
        ClaimType = table.Column<string>(nullable: true),
        ClaimValue = table.Column<string>(nullable: true)
    }

```

```

    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetRoleClaims", x => x.Id);
        table.ForeignKey(
            name: "FK_AspNetRoleClaims_AspNetRoles_RoleId",
            column: x => x.RoleId,
            principalTable: "AspNetRoles",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.CreateTable(
    name: "AspNetUserClaims",
    columns: table => new
    {
        Id = table.Column<int>(nullable: false)
            .Annotation("SqlServer:Identity", "1, 1"),
        UserId = table.Column<string>(nullable: false),
        ClaimType = table.Column<string>(nullable: true),
        ClaimValue = table.Column<string>(nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetUserClaims", x => x.Id);
        table.ForeignKey(
            name: "FK_AspNetUserClaims_AspNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.CreateTable(
    name: "AspNetUserLogins",
    columns: table => new
    {
        LoginProvider = table.Column<string>(nullable: false),
        ProviderKey = table.Column<string>(nullable: false),
        ProviderDisplayName = table.Column<string>(nullable: true),
        UserId = table.Column<string>(nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetUserLogins", x => new { x.LoginProvider,
x.ProviderKey });
        table.ForeignKey(
            name: "FK_AspNetUserLogins_AspNetUsers_UserId",

```

```

        column: x => x.UserId,
        principalTable: "AspNetUsers",
        principalColumn: "Id",
        onDelete: ReferentialAction.Cascade);
});

migrationBuilder.CreateTable(
    name: "AspNetUserRoles",
    columns: table => new
    {
        UserId = table.Column<string>(nullable: false),
        RoleId = table.Column<string>(nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetUserRoles", x => new { x.UserId, x.RoleId });
        table.ForeignKey(
            name: "FK_AspNetUserRoles_AspNetRoles_RoleId",
            column: x => x.RoleId,
            principalTable: "AspNetRoles",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
        table.ForeignKey(
            name: "FK_AspNetUserRoles_AspNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",
            principalColumn: "Id",
            onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.CreateTable(
    name: "AspNetUserTokens",
    columns: table => new
    {
        UserId = table.Column<string>(nullable: false),
        LoginProvider = table.Column<string>(nullable: false),
        Name = table.Column<string>(nullable: false),
        Value = table.Column<string>(nullable: true)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetUserTokens", x => new { x.UserId,
x.LoginProvider, x.Name });
        table.ForeignKey(
            name: "FK_AspNetUserTokens_AspNetUsers_UserId",
            column: x => x.UserId,
            principalTable: "AspNetUsers",
            principalColumn: "Id",

```

```

        onDelete: ReferentialAction.Cascade);
    });

migrationBuilder.InsertData(
    table: "AspNetRoles",
    columns: new[] { "Id", "ConcurrencyStamp", "Name", "NormalizedName" },
    values: new object[] { "44546e06-8719-4ad8-b88a-f271ae9d6eab", "1c07fb23-
ced7-48f9-bf0a-e0df233cd7a3", "admin", "ADMIN" });

migrationBuilder.InsertData(
    table: "AspNetUsers",
    columns: new[] { "Id", "AccessFailedCount", "ConcurrencyStamp", "Email",
"EmailConfirmed", "LockoutEnabled", "LockoutEnd", "NormalizedEmail",
"NormalizedUserName", "PasswordHash", "PhoneNumber", "PhoneNumberConfirmed",
"SecurityStamp", "TwoFactorEnabled", "UserName" },
    values: new object[] { "3b62472e-4f66-49fa-a20f-e7685b9565d8", 0, "091be5bf-
9d58-40e6-98a8-e82859894571", "my@email.com", true, false, null, "MY@EMAIL.COM",
"ADMIN",
"AQAAAAEAAcQAAAAECBA6mt3xGNgyLjwyRhhtl3PI2IBKsm00Y3bAJfdVrgT0++e45OV
5Vh4SLTLPDHEQ==", null, false, "", false, "admin" });

migrationBuilder.InsertData(
    table: "TextFields",
    columns: new[] { "Id", "CodeWord", "DateAdded", "MetaDescription",
"MetaKeywords", "MetaTitle", "Subtitle", "Text", "Title", "TitleImagePath" },
    values: new object[,]
    {
        { new Guid("63dc8fa6-07ae-4391-8916-e057f71239ce"), "PageIndex", new
DateTime(2020, 2, 8, 6, 13, 50, 96, DateTimeKind.Utc).AddTicks(9537), null, null, null, null,
"Содержание заполняется администратором", "Главная", null },
        { new Guid("70bf165a-700a-4156-91c0-e83fce0a277f"), "PageServices", new
DateTime(2020, 2, 8, 6, 13, 50, 97, DateTimeKind.Utc).AddTicks(2218), null, null, null, null,
"Содержание заполняется администратором", "Наши услуги", null },
        { new Guid("4aa76a4c-c59d-409a-84c1-06e6487a137a"), "PageContacts", new
DateTime(2020, 2, 8, 6, 13, 50, 97, DateTimeKind.Utc).AddTicks(2284), null, null, null, null,
"Содержание заполняется администратором", "Контакты", null }
    });

migrationBuilder.InsertData(
    table: "AspNetUserRoles",
    columns: new[] { "UserId", "RoleId" },
    values: new object[] { "3b62472e-4f66-49fa-a20f-e7685b9565d8", "44546e06-
8719-4ad8-b88a-f271ae9d6eab" });

migrationBuilder.CreateIndex(
    name: "IX_AspNetRoleClaims_RoleId",
    table: "AspNetRoleClaims",
    column: "RoleId");

```

```

migrationBuilder.CreateIndex(
    name: "RoleNameIndex",
    table: "AspNetRoles",
    column: "NormalizedName",
    unique: true,
    filter: "[NormalizedName] IS NOT NULL");

migrationBuilder.CreateIndex(
    name: "IX_AspNetUserClaims_UserId",
    table: "AspNetUserClaims",
    column: "UserId");

migrationBuilder.CreateIndex(
    name: "IX_AspNetUserLogins_UserId",
    table: "AspNetUserLogins",
    column: "UserId");

migrationBuilder.CreateIndex(
    name: "IX_AspNetUserRoles_RoleId",
    table: "AspNetUserRoles",
    column: "RoleId");

migrationBuilder.CreateIndex(
    name: "EmailIndex",
    table: "AspNetUsers",
    column: "NormalizedEmail");

migrationBuilder.CreateIndex(
    name: "UserNameIndex",
    table: "AspNetUsers",
    column: "NormalizedUserName",
    unique: true,
    filter: "[NormalizedUserName] IS NOT NULL");
}

protected override void Down(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropTable(
        name: "AspNetRoleClaims");

    migrationBuilder.DropTable(
        name: "AspNetUserClaims");

    migrationBuilder.DropTable(
        name: "AspNetUserLogins");

    migrationBuilder.DropTable(

```

```
        name: "AspNetUserRoles");

migrationBuilder.DropTable(
    name: "AspNetUserTokens");

migrationBuilder.DropTable(
    name: "ServiceItems");

migrationBuilder.DropTable(
    name: "TextFields");

migrationBuilder.DropTable(
    name: "AspNetRoles");

migrationBuilder.DropTable(
    name: "AspNetUsers");
    }
}
}
```